

Digital Terrain Elevation Data (DTED) Exporter
Technical Report

Tobi L.S. Sellekaerts

September 19, 1997

Prepared for:

7th Army Training Command (7th ATC)
Grafenwoehr, Germany

Terrain Modeling Project Office (TMPO)
Washington, D.C.

Under Contract:
DAJA 22-96-D0069

Prepared by:

Logicon, RDA
Grafenwoehr, Germany

In April of 1997, I wrote a DTED exporter in response to an exercise which needed DTED over an area where none existed: Zaire. The exporter uses as input an Arc GRID and outputs a single 1x1 degree properly formatted NIMA DTED level 1 file.

The exporter consists of three programs:

1. **griddted.aml** - an Arc/Info AML which takes a raster file of elevation values in Arc GRID format and converts it into a text file for processing.
2. **format_dted.perl** - a perl script which writes out all the header information for the DTED file and reformats the text file created by griddted.aml.
3. **asc2dted** - a binary executable, written by NIMA and obtained from their office in St. Louis, which formats the text file of elevations values and writes them out in the appropriate binary format. I spent a few weeks attempting to write my own version of asc2dted and felt I was very close to the solution when NIMA offered to send us their exporter so I didn't need to finish mine.

The AML should reside in the a tool folder, the other two programs in the u tool folder; these are both in \$ARCHOME. See Arc HELP for more information about placement of custom written AMLs and executables.

Other information about the exporter:

1. The input GRID cannot be bigger than 1 x 1 degree in size. The exporter could be modified to chop up GRIDs that are bigger but currently doesn't.
2. The input GRID can be in any projection, with any cell size, and does not need to cover the entire 1 x 1 degree area.
3. The output DTED cell will cover an entire 1 x 1 degree area, filling cells that don't have elevations values with the NIMA null value, -32767.
4. The area can be anywhere in the four quadrants (north or south, east or west hemisphere).
5. The exporter has been tested successfully with loading the output DTED into JTS and importing it into Arc/Info and ERDAS Imagine.
6. The exporter currently only creates Level 1, even if your input GRID is at a density comparable to a higher resolution of DTED. It could be modified to output other levels of DTED but currently doesn't.

The text of the exporter follows.

1. griddted.aml

```
&args grid

/* griddted.aml
/*
 * Program: griddted.aml
 * Purpose: duh... to export DTED. Right now this works in both
 * the northern and southern hemispheres, and at any latitude.
 * However, I haven't modified it to work with anything but
 * DTED level 1. So if you have amazingly dense data, it will
 * be downsampled to level 1!!!
/*
 * Called By: none
 * Calls Made: none
/*
 * Inputs: the name of the grid to export to DTED
 * Outputs: a correctly formatted DTED cell, or more than one
 * if necessary. Should work at all latitudes.
/*
 * History: 24 March 1997 - Tobi Steinberg
 * Terrain Simulations - 7th ATC - Grafenwoehr, Germany
 * Changed on 15 April 1997 to use GRIDASCII instead of DEM
/*
 * &severity &error &routine bail
/* &echo &on
display 0
/* &ty [date -VFULL]

/* Checking validity of entered arguments
&if [null %grid%] &then &do
    lg
    &ty Usage: GRIDDTED <grid>
    &return
&end

/* Shortening the name if it is too long
&if [length %grid%] gt 4 &then &s outgrid [substr %grid% 1 4]
&else &s outgrid %grid%
&s outgrid xx%outgrid%

/* Checking on the existence of grid names needed by the program
&do gridx &list %outgrid%_ds %outgrid%_ds2
    &if [exists %gridx% -grid] &then kill %gridx% all
&end

/* Checking on the existence of file names needed by the program
&do file &list %outgrid%_ds2.asc xperlstuff.txt
    &if [exists %file% -file] &then &sys rm %file%
&end

/* If the specifications of the grid projection don't match those
/* necessary for DTED, reproject.
&describe %grid%
&if %prj$name% ne GEOGRAPHIC OR %prj$units% ne DS OR ~
    %prj$spheroid% ne WGS84 &then &do
        /* using a variable to determine whether or not the input datum
        /* has been defined
```

```
&ty Projecting into geographic, with WGS84 spheroid and
&ty units of decimal seconds.
&s datdef 0
&if ^ [null %prj$datum%] &then &s datdef 1
project grid %grid% %outgrid%_ds
    output
        projection geographic
        units ds
        spheroid wgs84
        &if %datdef% = 1 &then datum wgs84
        parameters
            end
&end
&else copy %grid% %outgrid%_ds
&s grid %outgrid%

/* Setting a variable called resample so that the grid doesn't have to
/* be resampled more than once.
&s resample 0

/* Determining the correct cell size for the latitude of the area
&s ycell 3
&describe %grid%_ds
&s lat [truncate [calc %grd$xmin% / 3600]]
&if %lat% ge 80 &then &s xcell 18
&else &if %lat% ge 75 AND %lat% lt 80 &then &s xcell 12
    &else &if %lat% ge 70 AND %lat% lt 75 &then &s xcell 9
        &else &if %lat% ge 50 AND %lat% lt 70 &then &s xcell 6
            &else &s xcell 3

/* If the grid doesn't have the correct cell sizes, resample
/* it so it does.
&if %img$dx% ne %xcell% AND %img$dy% ne %ycell% &then &s resample 1

/* If the grid is larger than a 1 x 1 degree cell, resample
/* Need to change this part to account for overlap!!!!!!!!!!!!!
&if [calc %grd$xmax% - %grd$xmin%] ne 3603 AND ~
    [calc %grd$ymax% - %grd$ymin%] ne 3603 &then &s resample 1

/* If the grid doesn't cover an entire 1 x 1 degree cell, resample
/* it so that it does
/* &if [mod [truncate [calc [abs %grd$xmin%] + .5]] 3600] ne 3599 OR ~
/* [mod [truncate [calc [abs %grd$xmax%] - .5]] 3600] ne 1 OR ~
/* [mod [truncate [calc [abs %grd$ymin%] + .5]] 3600] ne 2 OR ~
/* [mod [truncate [calc [abs %grd$ymax%] - .5]] 3600] ne 3598 &then
/*     &s resample 1

/* Okay, now that all the tests are done, resample if necessary
&if %resample% = 1 &then &do
    &s truemin [calc [truncate [calc %grd$xmin% / 3600]] * 3600]
    &s truexmax [calc %truexmin% + 3600]
    &s trueymin [calc [truncate [calc %grd$ymin% / 3600]] * 3600]
    &s trueymax [calc %trueymin% + 3600]
    laticeresample %grid%_ds %grid%_ds2
        %truexmin%, %trueymin%
        %truexmax%, %trueymax%
        ~
        %xcell%, %ycell%
&end
```

```

&else copy %grid%_ds %grid%_ds2
&if [exists %grid%_ds2 -grid] &then kill %grid%_ds all

/* figuring out the geographic corners for use later and
/* reformatting them to fit the DTED naming convention
&if %grd$xmin% lt 0 &then &s grdxmin [calc %grd$xmin% - 2]
&else &s grdxmin [calc %grd$xmin% + 2]
&if %grd$ymin% lt 0 &then &s grdymin [calc %grd$ymin% - 2]
&else &s grdymin [calc %grd$ymin% + 2]
&s truexmin [calc [truncate [calc %grdxmin% / 3600]]]
&s trueymin [truncate [calc %grdymin% / 3600]]

&if %truexmin% lt 0 &then
  &s long w
&else &s long e
&s truexmin [abs %truexmin%]
&if %truexmin% lt 10 &then &s dtedx 0%truexmin%
&else &if %truexmin% lt 100 &then &s dtedx 0%truexmin%
  &else &s dtedx %truexmin%

&if %trueymin% lt 0 &then
  &s lat s
&else &s lat n
&s trueymin [abs %trueymin%]
&if %trueymin% lt 10 &then &s dtedy 0%trueymin%
&else &s dtedy %trueymin%

&s dtedname %long%dtedx%lat%dtedy%.dt1
&if [exists %dtedname% -file] &then &sys rm %dtedname%

/* export to an ascii file
gridascii %grid%_ds2 %grid%_ds2.asc

/* export to a lattice - changed 15 April 1997
/* latticedem %grid%_ds2 %grid%_ds2.dem # int

&describe %grid%_ds2
/* write variabes needed by the Perl script to a text file
&s toperl [open xperlstuff.txt os -write]
&s junk [write %toperl% %grid%_ds2.txt]
&s junk [write %toperl% %dtedname%]
&s junk [write %toperl% %xcell%]
&s junk [write %toperl% %ycell%]
&s junk [write %toperl% %grd$rows%]
&s junk [write %toperl% %grd$ncols%]
&s junk [write %toperl% [date -TAG]]
&s junk [close %toperl%]

/* run the perl script to reformat the text file to dted
format_dted.perl

/* post script activities
&if [exists x%dtedname% -file] &then &s junk [delete x%dtedname% -file]
&ty Printing elevation information...
&sys asc2dted %grid%_ds2.asc x%dtedname% TAPE01 0
&sys rm %grid%_ds2.*

/* clean up

```

```

&sys cat xx%dtedname% x%dtedname% > %dtedname%
&s junk [delete x%dtedname% -file]
&s junk [delete xx%dtedname% -file]
&s junk [delete xxdemtemp.txt -file]
&s junk [delete xxdtedtemp.dtl -file]
&s junk [delete xperlstuff.txt -file]
kill %grid%_ds2 all

/* &echo &off
display 9999
/* &ty [date -VFULL]
&return
*****&*****
/* bail routine
*****&*****
&routine bail
&severity &error &ignore
&severity &warning &ignore
&type An error has occured in export_dted.aml
&type Bailing out of export_dted.aml
&return; &return &error

```

2. format_dted.perl

```
#!/usr/local/bin/perl

# format_dted.perl
#
# Program: format_dted.perl
# Purpose: to format DEM data and a pre-made DTED header file
# into the appropriate DTED format
#
# Called By: export_dted.aml
# Calls Made: none
#
# Inputs: A file written by the AML listing the information
# described below; a DEM also made in ARC.
# Outputs: A complete, correctly formatted DTED file.
#
# History: 20-31 March 1997 - Tobi Steinberg
# steinbet@email.grafenwoehr.army.mil
# http://199.56.131.202/~tobi/
# Terrain Simulations - 7th ATC - Grafenwoehr, Germany
# Changed on 15 April 1997 to use GRIDASCII
#
# note - this works in both the northern and southern hemispheres, and
# at any latitude. However, it currently only works for DTED level 1.

# reading the perl information out of the text file created by the AML
open (PERLINFO, "xperlstuff.txt");
$demfile = <PERLINFO>; # the correct name of the output DTED file
$dtedfile = <PERLINFO>; # the name of the dem file to open for reading
$xcell = <PERLINFO>; # the x distance between elevation pts in ds
#(us. 3)
$ycell = <PERLINFO>; # the y distance between elevation pts in ds
#(us. 3)
$latrows = <PERLINFO>; # the number of rows of points (usually 1201)
$longcols = <PERLINFO>; # the number of columns (usually 1201)
$date6 = <PERLINFO>; # the date in YYMMDD format
close (PERLINFO);
print STDOUT "Exporting header information to $dtedfile...\n";

chop($latrows);
chop($longcols);
# opening the DEM and DTED files for reading and writing, respectively
open (DEMIN, "$demfile");
$temp = "xx" . "$dtedfile";
open (DTEDOUT, ">$temp");

# reading in, formatting, and printing out the header file information
# doing the extents first. Yes, this is kind of screwy, as I will have
# to do it all over again later, but necessary because of the southern
# and western hemispheres (why can't the south pole and -180 be the
# origin?)
$westx = substr ($dtedfile, 1, 3);
$southy = substr ($dtedfile, 5, 2);
$longhemi = substr ($dtedfile, 0, 1);
$lathemi = substr ($dtedfile, 4, 1);
```

```
# working on x
$westx *= 1;
if ($longhemi eq "w") {
    # $westx += 1;
    $westx *= -1;
}
$eastx = $westx + 1;
$westx = abs $westx;
$eastx = abs $eastx;

# working on y
$southy *= 1;
if ($lathemi eq "s") {
    # $southy += 1;
    $southy *= -1;
}
$northy = $southy + 1;
$southy = abs $southy;
$northy = abs $northy;

# now creating eastxf, westxf, southyf, northyf, which are the
# correctly
# formatted versions of these numbers for the DTED file
$lathemi =~ tr/a-z/A-Z/;
$longhemi =~ tr/a-z/A-Z/;
if ($eastx < 10) {
    $eastxf = "00" . "$eastx" . "0000" . "$longhemi";
} elsif ($eastx < 100) {
    $eastxf = "0" . "$eastx" . "0000" . "$longhemi";
} else {
    $eastxf = "$eastx" . "0000" . "$longhemi";
}

if ($westx < 10) {
    $westxf = "00" . "$westx" . "0000" . "$longhemi";
    $westxfdp = "00" . "$westx" . "0000.0" . "$longhemi";
} elsif ($westx < 100) {
    $westxf = "0" . "$westx" . "0000" . "$longhemi";
    $westxfdp = "0" . "$westx" . "0000.0" . "$longhemi";
} else {
    $westxf = "$westx" . "0000" . "$longhemi";
    $westxfdp = "$westx" . "0000.0" . "$longhemi";
}

if ($northy < 10) {
    $northyf = "0" . "$northy" . "0000" . "$lathemi";
} else {
    $northyf = "$northy" . "0000" . "$lathemi";
}

if ($southy < 10) {
    $southyf = "0" . "$southy" . "0000" . "$lathemi";
    $southyfdp = "0" . "$southy" . "0000.0" . "$lathemi";
} else {
    $southyf = "$southy" . "0000" . "$lathemi";
    $southyfdp = "$southy" . "0000.0" . "$lathemi";
}

$southyspec = "0" . "$southyf";
```

```

# formatting the xcell and ycell values
chop($xcell);
chop($ycell);
if ($xcell < 10) {
    $xcellf = "00" . "$xcell" . "0";
} else {
    $xcellf = "0" . "$xcell" . "0";
}
# note - this is assuming that we are only working with DTED level 1
$ycellf = "00" . "$ycell" . "0";

# formatting production year and day
$yymm = substr($date6, 0, 4);

# Writing data to the output DTED file

# ***** UHL
print DTEDOUT "UHL1$westxf$southyspec$xcellf$ycellf";
print DTEDOUT " NAU ";
printf DTEDOUT "%4g", $longcols;
printf DTEDOUT "%4g", $latrows;
print DTEDOUT "0" ;

# ***** DSIU
print DTEDOUT " DSIU ";
print DTEDOUT " DTED10000000000000000000
01A";
print DTEDOUT "000000000000USAREURO
MILP89020009604MSLWGS84";
print DTEDOUT " ";
print DTEDOUT "$yymm ";
print DTEDOUT "$southyfdp$westxfdp$southyf$westxf$northyf$westxf";
print DTEDOUT "$northyf$eastxf$southyf$eastxf";
print DTEDOUT "0000000.0";
print DTEDOUT "$ycellf$xcellf";
printf DTEDOUT "%4g", $latrows;
printf DTEDOUT "%4g", $longcols;
print DTEDOUT "00
print DTEDOUT " ";
# end of 101
print DTEDOUT " ";
print DTEDOUT " ";
# end of 100
print DTEDOUT " ";
print DTEDOUT " ";
print DTEDOUT " ";
print DTEDOUT " ";
# end of 156

# ***** ACC
print DTEDOUT "ACC NA NA NA NA
00";
print DTEDOUT " ";

```

```

print DTEDOUT " ";
print DTEDOUT " ";
print DTEDOUT " ";
print DTEDOUT " ";
# end of 500
print DTEDOUT " ";
# end of 1000
print DTEDOUT " ";
# end of 1500
print DTEDOUT " ";
# end of 2000
print DTEDOUT " ";
# end of 2500
print DTEDOUT " ";
print DTEDOUT " ";
print DTEDOUT " ";
# end of 2644
close (DEMIN);
close (DTEDOUT);


```

3. asc2dted

As **asc2dted** is a binary executable (more specifically, a “Sun demand paged SPARC executable dynamically linked”), it cannot be written here in ASCII.